සියලු ම හිමිකම් ඇවිරිණි / *All Rights Reserved*]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 234 MARKING

2025/08/01
ictfromabc
Ravindu Bandaranayake

1. What is the primary purpose of a USB (Universal Serial Bus) port?
   USB port එකක මූලික අරමුණ කුමක්ද?

   1) To provide power to the CPU / CPU එකට බලය සැපයීම.
   2) To connect the CPU to the motherboard / CPU එක මවු පුවරුවට සම්බන්ධ කිරීම.
   3) To connect storage devices like hard drives / දෘඩ තැටි වැනි ගබඩා උපාංග සම්බන්ධ කිරීම.
   4) To connect a wide variety of peripheral devices to a computer.
      පරිගණකයකට විවිධ පර්යන්ත උපාංග සම්බන්ධ කිරීම.
   5) To establish network connections between computers / පරිගණක අතර ජාල සම්බන්ධතා ඇති කිරීම.

   Answer : 4)

   Explanation /පැහැදිලි කිරීම

   The "Universal" part of the name is key. It was designed to standardize the connection of many different types of devices.
   නමෙහි "Universal" කොටස ප්‍රධාන වේ. එය විවිධ වර්ගයේ උපාංගවල සම්බන්ධතාවය ප්‍රමිතිකරණය කිරීම සඳහා නිර්මාණය කර ඇත.

   Peripheral Devices / පර්යන්ත උපාංග
   This includes things like keyboards, mice, printers, scanners, webcams, external hard drives, flash drives, and many more. USB provides both a standard data transfer interface and a power supply to these devices.
   යතුරුපුවරු, මූසික, මුද්‍රණ යන්ත්‍ර, ස්කෑනර්, වෙබ් කැමරා, බාහිර දෘඩ තැටි, ෆ්ලෑෂ් ධාවක සහ තවත් බොහෝ දේ මෙයට ඇතුළත් වේ. USB මඟින් මෙම උපාංග සඳහා සම්මත දත්ත හුවමාරු අතුරුමුහුණතක් සහ බල සැපයුමක් සපයයි.

   Therefore the correct answer is, 4).
   එබැවින් නිවැරදි පිළිතුර වන්නේ, 4) වේ.

2. What is the primary purpose of an arithmetic logic unit (ALU) in a CPU?
   CPU එකක ALU හි මූලික අරමුණ කුමක්ද?

   1) To store data permanently. / දත්ත ස්ථිරවම ගබඩා කිරීමට.
   2) To control the flow of instructions. / උපදෙස් ප්‍රවාහය පාලනය කිරීමට.
   3) To perform arithmetic and logical operations. / අංක ගණිත හා තාර්කික මෙහෙයුම් සිදු කිරීමට.
   4) To connect peripheral devices. / පර්යන්ත උපාංග සම්බන්ධ කිරීමට.
   5) To manage the computer's memory. / පරිගණකයේ මතකය කළමනාකරණය කිරීමට.

   Answer : 3)

   Explanation /පැහැදිලි කිරීම

   ALU stands for Arithmetic Logic Unit. The name itself explains its function.
   ALU යනු අංක ගණිත තාර්කික ඒකකයයි. එම නමෙන්ම එහි ක්‍රියාකාරිත්වය පැහැදිලි කෙරේ.

   Arithmetic operations include addition, subtraction, multiplication, and division.
   අංක ගණිත මෙහෙයුම් වලට එකතු කිරීම, අඩු කිරීම, ගුණ කිරීම සහ බෙදීම ඇතුළත් වේ.

Logical operations include comparisons (like greater than, less than, or equal to) and Boolean operations (AND, OR, NOT).
තාර්කික මෙහෙයුම් වලට සංසන්දනයන් (වඩා විශාල, අඩු හෝ සමාන වැනි) සහ බූලියන් මෙහෙයුම් (AND, OR, NOT) ඇතුළත් වේ.

The ALU is the "brain" of the CPU that actually performs these calculations and comparisons, which are the foundation of almost all computer tasks.
ALU යනු මෙම ගණනය කිරීම් සහ සංසන්දනයන් සැබවින්ම සිදු කරන CPU හි "brain" වන අතර එය පරිගණක කාර්යයන් සියල්ලේම පදනම වේ.

Therefore the correct answer is, 3).
එබැවින් නිවැරදි පිළිතුර වන්නේ, 3) වේ.

3. In 2's complement binary addition, how is the carry bit handled?
2 හි අනුපූරක ද්විමය එකතු කිරීමේදී, රැගෙන යන බිටුව හසුරුවන්නේ කෙසේද?

1) It is added back into the sum. / එය නැවත sum එකට එකතු කරනු ලැබේ.
2) It is stored separately for later use. / එය පසුව භාවිතා කිරීම සඳහා වෙන වෙනම ගබඩා කරනු ලැබේ.
3) <u>It is discarded. / එය ඉවත දමනු ලැබේ.</u>
4) It is inverted before being added to the sum. / sum එකට එකතු කිරීමට පෙර එය ප්‍රතිලෝම කරනු ලැබේ.
5) It affects the final sum. / එය අවසාන sum එකට බලපායි.

Answer : 3)

Explanation /පැහැදිලි කිරීම

In two's complement arithmetic, the carry-out from the most significant bit (the leftmost bit) is simply ignored.
දෙකේ අනුපූරක අංක ගණිතයේදී, වැඩිම වෙසෙසි බිටුවෙන් (වම් කෙළවරේ බිට් එක) රැගෙන යාම සරලව නොසලකා හරිනු ලැබේ.

The result of the addition is determined by the bits that fit within the fixed-width representation. This is because the two's complement system is a modular arithmetic system.
එකතු කිරීමේ ප්‍රතිඵලය තීරණය වන්නේ ස්ථාවර පළල නිරූපණය තුළට ගැලපෙන බිටු මගිනි. මෙයට හේතුව දෙකේ අනුපූරක පද්ධතිය මොඩියුලර් අංක ගණිත පද්ධතියක් වීමයි.

The carry-out is a natural consequence of this modular behavior and does not indicate an overflow on its own.
රැගෙන යාම මෙම මොඩියුලර් හැසිරීමේ ස්වාභාවික ප්‍රතිවිපාකයක් වන අතර එය තනිවම පිටාර ගැලීමක් නොපෙන්වයි.

An overflow condition is detected by checking if the carry-in to the most significant bit is different from the carry-out. If they are different, an overflow has occurred. If they are the same, no overflow has occurred, regardless of whether a carry-out was generated.
වැඩිම වෙසෙසි බිටුවට ගෙනයාම රැගෙන යාමෙන් වෙනස් දැයි පරීක්ෂා කිරීමෙන් පිටාර ගැලීමේ තත්වයක් අනාවරණය වේ. ඒවා වෙනස් නම්, පිටාර ගැලීමක් සිදුවී ඇත. ඒවා සමාන නම්, ගෙනයාමක් ජනනය කර තිබේද යන්න නොසලකා, පිටාර ගැලීමක් සිදුවී නොමැත.

Therefore the correct answer is, 3).
එබැවින් නිවැරදි පිළිතුර වන්නේ, 3) වේ.

4. Evaluate the following expression and give the result in decimal (base 10)
පහත ප්‍රකාශනය ඇගයීමට ලක් කර ප්‍රතිඵලය දශමයෙන් දෙන්න (පාදය 10)

$1101.101_2 + 2F_{16} - 34_8$

1) 32.625 2) 33.75 3) 34.25 4) 35.5 5) 30.5

Answer : 1)

Explanation/**පැහැදිලි කිරීම**

We start by converting each number from its base into decimal
අපි ආරම්භ කරන්නේ සෑම සංඛ්‍යාවක්ම එහි පාදයේ සිට දශමයට පරිවර්තනය කිරීමෙනි.

1101.101_2

	8	4	2	1	.5	.25	.125
Base 2	1	1	0	1	1	0	1
	8	4	0	1	.5	0	.125
Base 10	8+4+1+.5+.125=13.625						

$2F_{16}$

Base 16	2				F			
Base 2	0	0	1	0	1	1	1	1
Base 10	128	64	32	16	8	4	2	1
	32 +8+4+2+1 = 47							

34_8

Base 8	3			4		
Base 2	0	1	1	1	0	0
Base 10	32	16	8	4	2	1
	16+8+4 = 28					

Now we calculate the full expression in decimal
දැන් අපි සම්පූර්ණ ප්‍රකාශනය දශමයෙන් ගණනය කරමු.

= 13.625 +47 -28
=32.625

Therefore, the answer is 1)
එබැවින් පිළිතුර 1) වේ.

5. Simplify the following Boolean expression and select the correct answer.
පහත බූලියන් ප්‍රකාශනය සුළු කර නිවැරදි පිළිතුර තෝරන්න.

$(A + B).(\overline{A} + B).(A + \overline{B})$

1) A 2) B 3) A.B 4) A+B 5) 0

Answer : 3)

$(A + B).(\overline{A} + B).(A + \overline{B})$

This boolean expression can be simplfied as follows.
මෙම බූලියානු ප්‍රකාශනය පහත ආකාරයට සුළු කළ හැකිය.

$(\overline{A}+B)(A+\overline{B})A+(\overline{A}+B)(A+\overline{B})B$	Distributive Law විෂටන න්‍යාය
$(\overline{A}+B)A\overline{A}+(A+\overline{B})AB+(\overline{A}+B)(A+\overline{B})B$	Distributive Law විෂටන න්‍යාය
$0+(A+\overline{B})AB+(\overline{A}+B)(A+\overline{B})B$	Inverse Law ප්‍රතිලෝම න්‍යාය
$(A+\overline{B})AB+(\overline{A}+B)(A+\overline{B})B$	Identity Law සර්වසාම්‍ය න්‍යාය
$ABA+AB\overline{B}(\overline{A}+B)(A+\overline{B})B$	Distributive Law විෂටන න්‍යාය
$AB+AB\overline{B}(\overline{A}+B)(A+\overline{B})B$	Idempotent Law තද්ේවභාවී න්‍යාය
$AB+0+(\overline{A}+B)(A+\overline{B})B$	Inverse Law ප්‍රතිලෝම න්‍යාය
$AB+(\overline{A}+B)(A+\overline{B})B$	Identity Law සර්වසාම්‍ය න්‍යාය
$AB+(A+\overline{B})B\overline{A}+(A+\overline{B})BB$	Distributive Law විෂටන න්‍යාය
$AB+(A+\overline{B})B\overline{A}+(A+\overline{B})B$	Idempotent Law තද්ේවභාවී න්‍යාය
$AB+(A+\overline{B})B$	Redundancy Law සමතිරික්ත න්‍යාය
$AB+BA+B\overline{B}$	Distributive Law විෂටන න්‍යාය
$AB+BA+0$	Inverse Law ප්‍රතිලෝම න්‍යාය
$AB+BA$	Identity Law සර්වසාම්‍ය න්‍යාය
AB	Idempotent Law තද්ේවභාවී න්‍යාය

Therefore the correct answer is **3).**
එබැවින් නිවැරදි පිළිතුර 3) වේ.

6. what is the most simplified expression of $\overline{0}(\overline{C.1}+0.A)+\overline{B.1}$ of the followings?
පහත සඳහන් ප්‍රකාශන අතරින් $\overline{0}(\overline{C.1}+0.A)+\overline{B.1}$ හි වඩාත්ම සරල කළ ප්‍රකාශනය කුමක්ද?

1) $\overline{C}$ 2) 1 3) $C+\overline{B}$ 4) $\overline{C}.\overline{B}$ 5) $\underline{\overline{C}+\overline{B}}$

Answer : 5)

Explanation /**පැහැදිලි කිරීම**

A Boolean expression is obtained and then simplified using Boolean laws.
බූලියන් ප්‍රකාශනයක් ලබාගෙන පසුව බූලියන් නීති භාවිතයෙන් සරල කරනු ලැබේ

$F=\overline{0}(\overline{C.1}+0.A)+\overline{B.1}$
$=\overline{0}(\overline{C.1}+0.A)+\overline{B}+\overline{1}$ De morgan's Law/ $[\overline{(A+B)}=\overline{A}.\overline{B}]$ඩී මෝගන් න්‍යාය
$=\overline{0}(\overline{C}+\overline{1}+0.A)+\overline{B}+\overline{1}$ De morgan's Law/ $[\overline{(A+B)}=\overline{A}.\overline{B}]$ඩී මෝගන් න්‍යාය
$=1(\overline{C}+0+0.A)+\overline{B}+0$ $[\overline{0}=1,\overline{1}=0]$
$=1(\overline{C}+0+0.A)+\overline{B}$ Identity Law $[A+0=A]$සර්වසාම්‍ය න්‍යාය
$=1(\overline{C}+0)+\overline{B}$ Identity Law $[A.0=0]$සර්වසාම්‍ය න්‍යාය
$=1(\overline{C})+\overline{B}$ Identity Law $[A+0=A]$සර්වසාම්‍ය න්‍යාය
$=\overline{C}+\overline{B}$ Identity Law $[A.1=A]$සර්වසාම්‍ය න්‍යාය
$=\overline{C}+\overline{B}$

Therefore, the answer is 5)
එබැවින් පිළිතුර 5) වේ.

7. A student is designing a security door system that opens only when two specific conditions are met
ශිෂ්‍යයෙක් ආරක්ෂක දොර පද්ධතියක් නිර්මාණය කරමින් සිටින අතර එය විවෘත වන්නේ නිශ්චිත කොන්දේසි දෙකක් සපුරා ඇති විට පමණි

- A correct RFID tag is scanned (Input A = 1) / නිවැරදි RFID ටැගයක් ස්කෑන් කරනු ලැබේ
- A keypad PIN is entered correctly (Input B = 1)
යතුරු පුවරුවේ PIN අංකය නිවැරදිව ඇතුළත් කර ඇත

They want to build this using only a single type of gate from a standard logic gate IC (e.g., 7400 for NAND or 7402 for NOR).
ඔවුන්ට මෙය ගොඩනැගීමට අවශ්‍ය වන්නේ සම්මත තාර්කික ද්වාර එකකින් (උදා: 7400 for NAND or 7402 for NOR) එක් ආකාරයක ද්වාරයක් පමණක් භාවිතා කරමිනි.

Which approach allows them to implement this logic using a single IC?
තනි IC එකක් භාවිතයෙන් මෙම තර්කනය ක්‍රියාත්මක කිරීමට ඔවුන්ට ඉඩ සලසන ප්‍රවේශය කුමක්ද?

1) Use a 7402 IC to build an AND gate by combining NOR gates
NOR ගේට්ටු ඒකාබද්ධ කිරීමෙන් AND ගේට්ටුවක් තැනීමට 7402 IC එකක් භාවිතා කරන්න.
2) Use a 7400 IC to build an AND gate by combining NAND gates
NAND gates ඒකාබද්ධ කිරීමෙන් AND ද්වාරයක් තැනීමට 7400 IC එකක් භාවිතා කරන්න.
3) Use a 7402 IC directly to create OR logic
OR තර්කනය නිර්මාණය කිරීමට කෙලින්ම 7402 IC භාවිතා කරන්න.
4) Use a 7400 IC directly to invert one input
එක් ආදානයක් ප්‍රතිලෝම කිරීමට 7400 IC එකක් කෙලින්ම භාවිතා කරන්න.
5) Use both 7400 and 7402 ICs to build the logic
තර්කනය ගොඩනැගීමට 7400 සහ 7402 ICs දෙකම භාවිතා කරන්න.

Answer : 2)

Explanation /පැහැදිලි කිරීම

The student needs to allow the door to open only when both conditions are true. That is, they need AND logic
ශිෂ්‍යයා දොර විවෘත කිරීමට ඉඩ දිය යුත්තේ කොන්දේසි දෙකම සත්‍ය වූ විට පමණි. එනම්, ඔවුන්ට සහ තර්කනය අවශ්‍ය වේ.

Option 1
While technically possible, building an AND gate using only NOR gates is less efficient than using NAND gates. You would need at least three NOR gates to implement an AND gate .
තාක්ෂණිකව කළ හැකි වුවද, NOR ගේට්ටු පමණක් භාවිතා කර AND ගේට්ටුවක් තැනීම NAND ගේට්ටු භාවිතා කිරීමට වඩා අඩු කාර්යක්ෂම වේ. AND ගේට්ටුවක් ක්‍රියාත්මක කිරීමට ඔබට අවම වශයෙන් NOR ගේට්ටු තුනක් අවශ්‍ය වනු ඇත.
This makes it inefficient and not ideal when using limited IC resources.
මෙය සීමිත IC සම්පත් භාවිතා කරන විට එය අකාර්යක්ෂම වන අතර පරමාදර්ශී නොවේ.

Option 2
The 7400 IC contains four NAND gates, and NAND is a universal gate. By wiring two NAND gates in a specific way, you can make them behave like an AND gate
7400 IC එකේ NAND ගේට්ටු හතරක් අඩංගු වන අතර NAND යනු විශ්වීය ගේට්ටුවකි. NAND ගේට්ටු දෙකක් නිශ්චිත ආකාරයකින් රැහැන්ගත කිරීමෙන්, ඔබට ඒවා AND ගේට්ටුවක් මෙන් හැසිරීමට සැලැස්විය හැකිය.

Option 3
This completely misses the required logic. The door should open only when both A and B are true. That's AND logic, not OR.
මෙය අවශ්‍ය තර්කනය සම්පූර්ණයෙන්ම මග හරියි. දොර විවෘත විය යුත්තේ A සහ B දෙකම සත්‍ය වූ විට පමණි. ඒ තර්කනය OR , AND නොවේ.

Option 4
This option suggests using the 7400 NAND IC as a NOT gate by connecting both inputs together. While this is possible, it only gives you a NOT function, not the full AND logic required.
මෙම විකල්පය මඟින් ආදාන දෙකම එකට සම්බන්ධ කිරීමෙන් 7400 NAND IC එක NOT ද්වාරයක් ලෙස භාවිතා කිරීමට යෝජනා කරයි. මෙය කළ හැකි වුවද, එය ඔබට ලබා දෙන්නේ NOT ශ්‍රිතයක් පමණි, අවශ්‍ය සම්පූර්ණ AND තර්කනය නොවේ.

Option 5
This option violates the problem's constraint. The scenario clearly states that the student can use only a single type of logic gate IC due to hardware or design limitations.
මෙම විකල්පය ගැටලුවේ සීමාව උල්ලංඝනය කරයි. දෘඩාංග හෝ සැලසුම් සීමාවන් හේතුවෙන් ශිෂ්‍යයාට භාවිතා කළ හැක්කේ එක් තාර්කික ද්වාර IC වර්ගයක් පමණක් බව සිද්ධියේ පැහැදිලිව සඳහන් වේ.

Therefore, the answer is 2)
එබැවින් පිළිතුර 2) වේ.

8. What does the `pop()` method do in Python?
Python හි `pop()` ක්‍රමය කරන්නේ කුමක්ද?

1) Adds an element to the end of a list. / ලැයිස්තුවක අවසානයට මූලාංගයක් එක් කරයි.
2) Removes an element from a list at a specified index.
නිශ්චිත දර්ශකයක ලැයිස්තුවකින් මූලාංගයක් ඉවත් කරයි.
3) Reverses the elements of a list. / ලැයිස්තුවක මූලාංග ප්‍රතිලෝම කරයි.
4) Sorts the elements of a list. / ලැයිස්තුවක මූලාංග වර්ග කරයි.
5) Appends a list to the end of another list. / ලැයිස්තුවක් වෙනත් ලැයිස්තුවක අවසානයට එකතු කරයි.

Answer : 2)

Explanation /පැහැදිලි කිරීම

The `pop()` method is used on lists.
ලැයිස්තු වල `pop()` ක්‍රමය භාවිතා වේ.

It removes and returns the element at a given index.
එය දී ඇති දර්ශකයක මූලාංග ඉවත් කර ආපසු ලබා දෙයි.

If no index is specified, it removes and returns the last element of the list.
දර්ශකයක් නිශ්චිතව දක්වා නොමැති නම්, එය ලැයිස්තුවේ අවසාන මූලාංග ඉවත් කර ආපසු ලබා දෙයි.

Therefore the correct answer is, 2).
එබැවින් නිවැරදි පිළිතුර වන්නේ, 2) වේ.

9. What is the output of the below Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
y = 5
for i in range(y):
   if i % 2 != 0:
     print(i, end=' ')
```

1) `1 2 3 4 5` 2) `0 1 2 3 4` 3) `1 3`
4) `0 2 4` 5) `1 3 5`

Answer : 3)

Explanation /**පැහැදිලි කිරීම**

`y = 5`
The variable **`y`** is assigned the value 5.
`y` **විචල්‍යයට** `5` **අගය පවරනු ලැබේ.**

`for i in range(y):`
This creates a loop that will iterate through the numbers from 0 up to (but not including) **`y`** . Since **`y`** is 5, the values of **`i`** will be 0, 1, 2, 3, and 4.
මෙය 0 **සිට** `y` **දක්වා සංඛ්‍යා හරහා පුනරාවර්තනය වන ලූපයක් නිර්මාණය කරයි.** `y` 5 **වන බැවින්,** `i` **හි අගයන්** 0, 1, 2, 3 **සහ** 4 **වනු ඇත.**

`if i % 2 != 0:`
Inside the loop, this **`if`** statement checks if the current value of **`i`** has a remainder when divided by 2. This is the condition for a number being odd.
ලූපය තුළ, මෙම `if` **ප්‍රකාශය** `i` **හි වත්මන් අගය** 2 **න් බෙදූ විට ඉතිරියක් තිබේද යන්න පරීක්ෂා කරයි. මෙය සංඛ්‍යාවක් ඔත්තේ නම් කොන්දේසියයි.**

`print(i, end=' ')`
This line only executes when the **`if`** condition is true. It prints the value of **`i`** , followed by a space instead of a new line.
මෙම රේඛාව ක්‍රියාත්මක වන්නේ `if` **කොන්දේසිය සත්‍ය වූ විට පමණි. එය** `i` **හි අගය මුද්‍රණය කරන අතර, පසුව නව රේඛාවක් වෙනුවට ඉඩක් ඇත.**

`i = 0: 0 % 2` is 0. The condition **`0 != 0`** is false. Nothing is printed.
`i = 1: 1 % 2` is 1. The condition **`1 != 0`** is true. **`1`** is printed.
`i = 2: 2 % 2` is 0. The condition **`0 != 0`** is false. Nothing is printed.
`i = 3: 3 % 2` is 1. The condition **`1 != 0`** is true. **`3`** is printed.
`i = 4: 4 % 2` is 0. The condition **`0 != 0`** is false. Nothing is printed.

Therefore the correct answer is, 3).
එබැවින් නිවැරදි පිළිතුර වන්නේ, 3) **වේ.**

10. What is the output of the following python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
def process(data):
    product = 1
    for x in data:
        if x % 2 == 1 and x > 5:
            product *= x
    return product

nums = [1, 3, 5, 7, 9, 11]
print(process(nums))
```

1) 693 2) 105 3) 231 4) 99 5) 0

Answer : 1)

Explanation /පැහැදිලි කිරීම

This Python program defines a function called process, which takes a list of numbers and calculates the product of all odd numbers that are greater than 5.
මෙම පයිතන් වැඩසටහන මඟින් ක්‍රියාවලිය ලෙස හඳුන්වන ශ්‍රිතයක් නිර්වචනය කරයි, එය සංඛ්‍යා ලැයිස්තුවක් ගෙන 5ට වඩා වැඩි සියලුම ඔත්තේ සංඛ්‍යා වල ගුණිතය ගණනය කරයි.

It starts by initializing a variable product to 1.
එය ආරම්භ වන්නේ විචල්‍ය නිෂ්පාදනයක් 1ට ආරම්භ කිරීමෙනි.

Then, it loops through each number in the input list and checks two conditions, whether the number is odd (x % 2 == 1) and whether it is greater than 5.
ඉන්පසු, එය ආදාන ලැයිස්තුවේ ඇති සෑම අංකයක්ම ලූප් කර කොන්දේසි දෙකක් පරීක්ෂා කරයි, අංකය ඔත්තේ ද (x % 2 == 1) සහ එය 5ට වඩා වැඩි ද යන්න.

If both conditions are met, the number is multiplied into the product.
කොන්දේසි දෙකම සපුරා ඇත්නම්, එම සංඛ්‍යාව ගුණිතයට ගුණ කරනු ලැබේ.

For the given list [1, 3, 5, 7, 9, 11], only the numbers 7, 9, and 11 satisfy both conditions.
දී ඇති ලැයිස්තුවේ [1, 3, 5, 7, 9, 11] කොන්දේසි දෙකම සපුරාලන්නේ 7, 9 සහ 11 යන සංඛ්‍යා පමණි.

Their product is calculated step by step: 7 × 9 = 63, and then 63 × 11 = 693.
ඔවුන්ගේ නිෂ්පාදිතය පියවරෙන් පියවර ගණනය කෙරේ: 7 × 9 = 63 පසුව 63 × 11 = 693

Therefore, the output of the program is 693.
එබැවින්, වැඩසටහනේ ප්‍රතිදානය 693 වේ

Therefore, the answer is 1) /එබැවින් පිළිතුර 1) වේ.